II. AMENDMENTS TO THE CLAIMS

The following is a listing of claims to replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for testing a server application using a reentrant test application, comprising:

providing a test application that satisfies reentrancy requirements on a client;

identifying application protocol interfaces (APIs) associated with the test application;

providing a test script capable of invoking the APIs; and

instantiating, via the test script, a plurality of instances of the test application using threads, wherein the instantiating and execution of each of the plurality of instances of the test application occur within a single process, sharing all services and memory space with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within.

2. (Previously Presented) The method of claim 1, wherein upon execution, the test script instantiates the plurality of instances of the test application using threads within the single process.

3. (Original) The method of claim 1, wherein the server application is a network application.

4. (Original) The method of claim 1, wherein the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other.

5. (Original) The method of claim 1, wherein each of the plurality of instances of the test application corresponds to a separate thread, and wherein each of the separate threads is associated with a different connection to the server.

6. (Previously Presented) The method of claim 1, wherein the process comprises a JAVA™ virtual machine.

7. (Original) The method of claim 1, wherein the plurality of instances of the test application simulate use of the server application by a plurality of users.

8. (Original) The method of claim 1, further comprising collecting data corresponding to the test.

9. (Currently Amended) A system for testing a server application, comprising

an interface identification system for identifying application protocol interfaces (APIs) associated with a test application; and

an application instantiation system for instantiating a plurality of instances of the test application on a client using threads if the test application satisfies reentrancy requirements, wherein the test application instantiation system comprises a driver that executes a test script capable of invoking the identified APIs, and wherein the instantiating and execution of each of the plurality of instances of the test application, via the test script, occur within a single process, sharing all services and memory space with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within.

10. (Cancelled)

11. (Previously Presented) The system of claim 9, wherein upon execution, the test script instantiates the plurality of instances of the test application using threads within the single process.

12. (Original) The system of claim 9, wherein the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other.

13. (Original) The system of claim 9, wherein each of the plurality of instances of the test application corresponds to a separate thread, and wherein each of the separate threads is associated with a different connection to the server.

14. (Original) The system of claim 9, wherein the application is a network application.

15. (Previously Presented) The system of claim 9, wherein the process comprises a JAVA™ virtual machine.

16. (Original) The system of claim 9, wherein the plurality of instances of the test application simulate use of the server application by a plurality of users.

17. (Original) The system of claim 9, further comprising a data collection system for collecting data corresponding to the test.

18. (Currently Amended) A program product stored on a recordable medium for testing a server application, which when executed, comprises

program code for identifying application protocol interfaces (APIs) associated with the test application;

program code for instantiating a plurality of instances of a test application on a client using threads if the test application satisfies reentrancy requirements, wherein the program code for instantiating executes a test script capable of invoking the identified APIs, and wherein the instantiating and execution of each of the plurality of instances of the test application, via the test script, occur within a single process, sharing all services and memory space with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within.

19. (Cancelled)

20. (Previously Presented) The program product of claim 18, wherein upon execution, the test script instantiates the plurality of instances of the test application using threads within the single process.

21. (Original) The program product of claim 18, wherein the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other.

22. (Original) The program product of claim 18, wherein each of the plurality of instances of the test application corresponds to a separate thread, and wherein each of the separates threads is associated with a different connection to the server.

23. (Original) The program product of claim 18, wherein the server application is a network application.

24. (Previously Presented) The program product of claim 18, wherein the process comprises a JAVA™ virtual machine.

25. (Original) The program product of claim 18, wherein the plurality of instances of the test application simulate use of the server application by a plurality of users.

26. (Original) The program product of claim 18, further comprising program code for collecting data corresponding to the test.